

# Analysis of Distributed ADMM Algorithm for Consensus Optimization in Presence of Node Error

Layla Majzoobi<sup>+</sup>, Farshad Lahouti<sup>\*</sup>, and Vahid Shah-Mansouri<sup>+</sup>  
<sup>+</sup>School of Electrical and Computer Engineering, University of Tehran  
<sup>\*</sup>Electrical Engineering Department, California Institute of Technology

## Abstract

Alternating Direction Method of Multipliers (ADMM) is a popular convex optimization algorithm, which can be employed for solving distributed consensus optimization problems. In this setting agents locally estimate the optimal solution of an optimization problem and exchange messages with their neighbors over a connected network. The distributed algorithms are typically exposed to different types of errors in practice, e.g., due to quantization or communication noise or loss. We here focus on analyzing the convergence of distributed ADMM for consensus optimization in presence of additive random node error, in which case, the nodes communicate a noisy version of their latest estimate of the solution to their neighbors in each iteration. We present analytical upper and lower bounds on the mean squared steady state error of the algorithm in case that the local objective functions are strongly convex and have Lipschitz continuous gradients. In addition we show that, when the local objective functions are convex and the additive node error is bounded, the estimation error of the noisy ADMM for consensus optimization is also bounded. Numerical results are provided which demonstrate the effectiveness of the presented analyses and shed light on the role of the system and network parameters on performance.

## Index Terms

ADMM algorithm, Consensus optimization, Convergence, Node error, Steady State Error.

## I. INTRODUCTION

In the advent of big data, the Internet of Things and cyber-physical systems, distributed processing algorithms have attracted increasing research attention. Considerable problems in these areas including network flow control, feature extraction and power system state estimation, are formulated as convex optimization problems, and use distributed optimization methods [2]- [9]. These algorithms may be characterized by their range of applicability, convergence behavior, performance, and computational complexity.

Alternating Direction Method of Multipliers (ADMM) is one of the popular convex optimization algorithms which can be implemented in a distributed manner. Wide range of applicability and convergence with an adequate accuracy within a few tens of iterations make ADMM very useful in practice. Different applications and convergence behavior of ADMM algorithm are discussed in [10] - [20]. An upper bound on the global linear convergence rate of ADMM algorithm for strongly convex objective function under smoothness assumptions is provided in [20]. A parameter selection method which optimizes the derived convergence bound is also presented in this work.

Decentralized consensus optimization is an important family of optimization problems which is formulated as

$$\min_{\tilde{\mathbf{x}}} \sum_{i=1}^N f_i(\tilde{\mathbf{x}}) \quad (1)$$

where  $\tilde{\mathbf{x}} \in \mathbb{R}^n$  and  $f_i(\tilde{\mathbf{x}}) : \mathbb{R}^n \rightarrow \mathbb{R}$  are convex functions.  $N$  agents aim to optimize the sum of local objective functions,  $f_i(\tilde{\mathbf{x}})$ ,  $i \in \{1, \dots, N\}$ , over a global variable  $\tilde{\mathbf{x}}$ . Distributed ADMM-based algorithms for consensus optimization problems can be classified into two categories. In the first one, there is a central

collector or fusion center that all agents communicate with. Indeed the agents and the fusion center form a star network [10]. In the second one, there is not a fusion center, and the underlying network between the agents can have any connected topology [21]–[24]. While the former category has better convergence rate, the flexibility of network topology in the latter one is more appealing. A framework is provided in [25], which improves the convergence property of the latter category by allowing multiple fusion centers in the network.

The distributed optimization algorithms rely on local computations and communication between neighbors. They potentially suffer from different types of errors and imperfections such as computation, quantization and communication errors and link failure which could substantially affect the performance of distributed optimization methods. The subgradient-push algorithm which is convergent over time varying networks is proposed in [26]. ADMM-based algorithms which are resilient to link failure are proposed in [27]. Certain local computations are too complex to be carried out exactly and are usually replaced by approximations. Inexact variants of ADMM algorithm are proposed in [21], [28]–[30]. In [21], three distributed ADMM-based algorithms are proposed to solve the Lasso problem offering different computational complexity and convergence rate trade-offs. In [28] each ADMM update step, which includes solving an optimization subproblem, is replaced by an inexact step which reduces the computational cost of the algorithm by an order of magnitude. In [29] and [30], the Lagrangian minimization in each ADMM step is, respectively, replaced by linear and quadratic approximations of the objective function. The quadratic approximation is more complex than the linear approximation, but results in an algorithm with better convergence properties. A distributed subgradient method for consensus optimization which is convergent in presence of quantization error is proposed in [31]. In [32], ADMM-based distributed algorithms are proposed for parameter estimation in ad hoc wireless sensor networks which exhibit resilience in the presence of communication/quantization noise. For the least square objective function, the steady state error of the algorithm is analyzed. A robust ADMM-based distributed algorithm for average consensus problem in presence of noise is presented in [33].

In this work, we focus on analyzing the convergence behavior of distributed ADMM for consensus optimization over a connected network in presence of additive node error. We consider general convex or strongly convex local objective functions. In this setting, the latest estimate of the solution computed at each node is subject to additive random noise prior to transmission to neighboring nodes (this for instance could be due to quantization error [34]). We provide analytical upper and lower bounds on the mean squared steady state error of the noisy ADMM algorithm for consensus, when local objective functions are strongly convex and have Lipschitz continuous gradients. In addition, we show that for convex, proper and closed local objective functions and in presence of bounded node error, the steady state error of noisy ADMM algorithm for consensus is bounded. Extensive numerical results are provided which shows the effectiveness of the presented analyses. We also study the effect of different factors, such as noise variance, network topology and ADMM algorithm parameter on the provided bounds. We also propose a method to tune the algorithm parameter, which results in a faster convergence rate and a smaller steady state error.

The rest of the paper is organized as follows. The system model and prior art is reviewed in good details in Section II. Section III analyzes the convergence behavior of the algorithm in presence of additive node error. Analytical results are numerically assessed in Section IV, and conclusions are drawn in Section V.

## II. PRELIMINARIES

Consider a set of  $N$  nodes that intend to solve the consensus optimization problem in (1) in a distributed manner and over a given connected network with  $E$  undirected links. The network can be represented by the graph  $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ , where  $\mathcal{V}$  is the set of nodes,  $|\mathcal{V}| = N$ , and  $\mathcal{A}$  is the set of arcs,  $|\mathcal{A}| = 2E$ . Communications of nodes are synchronous and restricted to the neighbors.

---

**Algorithm 1** : Distributed ADMM-based algorithm for consensus optimization problem

---

Input functions  $f_i$ ; Initialization: for all  $i \in \mathcal{V}$ , set  $\mathbf{x}_i^0 = \boldsymbol{\alpha}_i^0 = \mathbf{0}_{n \times 1} \in \mathbb{R}^n$ ; Set algorithm parameter  $c > 0$ ;  $k = 0$ ;

**for all**  $k = 0, 1, \dots$ , every node  $i$  **do**

1. Update  $\mathbf{x}_i^{k+1}$  by solving  $\nabla f_i(\mathbf{x}_i^{k+1}) + \boldsymbol{\alpha}_i^k + 2c|\mathcal{N}_i|\mathbf{x}_i^{k+1} - c(|\mathcal{N}_i|\mathbf{x}_i^k + \sum_{j \in \mathcal{N}_i} \mathbf{x}_j^k) = \mathbf{0}$  ;
  2. Node  $i$  sends  $\mathbf{x}_i^{k+1}$  to its neighbors;
  3. Update  $\boldsymbol{\alpha}_i^{k+1} = \boldsymbol{\alpha}_i^k + c(|\mathcal{N}_i|\mathbf{x}_i^{k+1} - \sum_{j \in \mathcal{N}_i} \mathbf{x}_j^{k+1})$ ;
- 

In this section, we briefly review the ADMM algorithm and the known convergence results from [24] for completeness. To apply the ADMM algorithm to problem in (1), we reformulate the problem as

$$\begin{aligned} \min_{\mathbf{x}_i, \mathbf{z}_{ij}} \quad & \sum_{i=1}^N f_i(\mathbf{x}_i) \\ \text{s.t.} \quad & \mathbf{x}_i = \mathbf{z}_{ij}, \quad \mathbf{x}_j = \mathbf{z}_{ij} \quad \forall (i, j) \in \mathcal{A} \end{aligned} \quad (2)$$

where  $\mathbf{x}_i \in \mathbb{R}^n$  is the copy of the global optimization variable  $\tilde{\mathbf{x}} \in \mathbb{R}^n$  at node  $i$  and  $\mathbf{z}_{ij} \in \mathbb{R}^n$  is the variable, which enforces equality of  $\mathbf{x}_i$  and  $\mathbf{x}_j$  at the optimal point. Concatenating  $\mathbf{x}_i$ s and  $\mathbf{z}_{ij}$ s respectively into  $\mathbf{x} \in \mathbb{R}^{nN}$  and  $\mathbf{z} \in \mathbb{R}^{2nE}$ , (2) can be rewritten in the following form

$$\min_{\mathbf{x}, \mathbf{z}} \quad f(\mathbf{x}) + g(\mathbf{z}); \quad \text{s.t.} \quad \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} = \mathbf{0} \quad (3)$$

where  $f(\mathbf{x}) = \sum_{i=1}^N f_i(\mathbf{x}_i)$ ,  $g(\mathbf{z}) = 0$ . Matrix  $\mathbf{A}$  can be partitioned as  $\mathbf{A} = [\mathbf{A}_1; \mathbf{A}_2]$ , where  $\mathbf{A}_1 = \hat{\mathbf{A}}_1 \otimes \mathbf{I}_n$  and  $\mathbf{A}_2 = \hat{\mathbf{A}}_2 \otimes \mathbf{I}_n$ , and  $\otimes$  represents the Kronecker product. Elements of  $\hat{\mathbf{A}}_1$  and  $\hat{\mathbf{A}}_2$  are zero or one. Assuming that  $(i, j) \in \mathcal{A}$  is an arc in the network graph, the  $(m, i)$  element of  $\hat{\mathbf{A}}_1$  and the  $(m, j)$  element of  $\hat{\mathbf{A}}_2$  are one if  $\mathbf{z}_{ij}$  is the  $m^{\text{th}}$  block of  $\mathbf{z}$ ,  $\forall m \in [1, 2E]$ . Also, we have  $\mathbf{B} = [-\mathbf{I}_{2nE}; -\mathbf{I}_{2nE}]$ . Forming augmented Lagrangian and applying ADMM algorithm, we have

$$\nabla f(\mathbf{x}^{k+1}) + \mathbf{A}^T \boldsymbol{\lambda}^k + c\mathbf{A}^T(\mathbf{A}\mathbf{x}^{k+1} + \mathbf{B}\mathbf{z}^k) = \mathbf{0} \quad (4a)$$

$$\mathbf{B}^T \boldsymbol{\lambda}^k + c\mathbf{B}^T(\mathbf{A}\mathbf{x}^{k+1} + \mathbf{B}\mathbf{z}^{k+1}) = \mathbf{0} \quad (4b)$$

$$\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k - c(\mathbf{A}\mathbf{x}^{k+1} + \mathbf{B}\mathbf{z}^{k+1}) = \mathbf{0} \quad (4c)$$

where  $\boldsymbol{\lambda}$  and  $c$  are Lagrange multiplier and ADMM parameter, respectively;  $\nabla f(\mathbf{x}^{k+1})$  is gradient or subgradient of  $f(\mathbf{x})$  at  $\mathbf{x}^{k+1}$  depending on the differentiability of  $f$ . Considering  $\boldsymbol{\lambda} = [\boldsymbol{\beta}; \boldsymbol{\nu}]$  with  $\boldsymbol{\beta}, \boldsymbol{\nu} \in \mathbb{R}^{2nE}$ ,  $\mathbf{M}_+ = \mathbf{A}_1^T + \mathbf{A}_2^T$  and  $\mathbf{M}_- = \mathbf{A}_1^T - \mathbf{A}_2^T$ , it can be shown that  $\boldsymbol{\beta} = -\boldsymbol{\nu}$  and with some algebraic manipulations [24], we have

$$\nabla f(\mathbf{x}^{k+1}) + \mathbf{M}_- \boldsymbol{\beta}^{k+1} - c\mathbf{M}_+(\mathbf{z}^k - \mathbf{z}^{k+1}) = \mathbf{0} \quad (5a)$$

$$\boldsymbol{\beta}^{k+1} - \boldsymbol{\beta}^k - \frac{c}{2}\mathbf{M}_-^T \mathbf{x}^{k+1} = \mathbf{0} \quad (5b)$$

$$\frac{1}{2}\mathbf{M}_+^T \mathbf{x}^{k+1} - \mathbf{z}^{k+1} = \mathbf{0} \quad (5c)$$

Multiplying (5b) by  $\mathbf{M}_-$  and substituting the result in (5a), and also replacing (5c) in (5a), we have

$$\nabla f(\mathbf{x}^{k+1}) + \mathbf{M}_- \boldsymbol{\beta}^k + 2c\mathbf{W}\mathbf{x}^{k+1} - \frac{c}{2}\mathbf{M}_+ \mathbf{M}_+^T \mathbf{x}^k = \mathbf{0} \quad (6a)$$

$$\beta^{k+1} - \beta^k - \frac{c}{2} \mathbf{M}_-^T \mathbf{x}^{k+1} = \mathbf{0}, \quad (6b)$$

where

$$\mathbf{W} = \frac{1}{4} (\mathbf{M}_+ \mathbf{M}_+^T + \mathbf{M}_- \mathbf{M}_-^T), \quad (7)$$

is the extended degree matrix of the underlying network. By “extended”, we mean replacing every 1 by  $\mathbf{I}_n$ , and 0 by  $\mathbf{0}_n$  in the original definition of this matrix [24]. Multiplying two sides of (6b) by  $\mathbf{M}_-$ , and defining  $\boldsymbol{\alpha} = [\boldsymbol{\alpha}_1; \boldsymbol{\alpha}_2; \dots; \boldsymbol{\alpha}_N] = \mathbf{M}_- \boldsymbol{\beta} \in \mathbb{R}^{nN}$ , a relatively simple fully decentralized algorithm is obtained in [24] and presented in Algorithm 1. In this Algorithm,  $\mathcal{N}_i$  is the set of neighbors of node  $i$ .

It is shown that with the following assumption, the proposed algorithm converges R-linearly to its optimal point over a fixed connected underlying network [24].

**Assumption 1.** The local objective functions are strongly convex and have Lipschitz continuous gradients. For each agent  $i$ , and given any  $\mathbf{x}_a, \mathbf{x}_b \in \mathbb{R}^n$ , we have

$$\begin{aligned} \langle \nabla f_i(\mathbf{x}_a) - \nabla f_i(\mathbf{x}_b), \mathbf{x}_a - \mathbf{x}_b \rangle &\geq m_{f_i} \|\mathbf{x}_a - \mathbf{x}_b\|_2^2, \\ \|\nabla f_i(\mathbf{x}_a) - \nabla f_i(\mathbf{x}_b)\|_2 &\leq M_{f_i} \|\mathbf{x}_a - \mathbf{x}_b\|_2, \end{aligned}$$

where  $m_{f_i} > 0$  and  $M_{f_i} > 0$ .

According to Assumption 1,  $f(\mathbf{x})$  is a strongly convex function with module  $m_f = \min_i m_{f_i}$  and its gradient is Lipschitz continuous with constant  $M_f = \max_i M_{f_i}$ .

**Theorem 1.** Consider the ADMM iterations in (5), that solves the optimization problem in (3). The primal variables  $\mathbf{x}$  and  $\mathbf{z}$ , respectively have their optimal values  $\mathbf{x}^*$  and  $\mathbf{z}^*$ , and the optimal value for the dual variable  $\boldsymbol{\beta}$  is  $\boldsymbol{\beta}^*$ . Considering

$$\mathbf{u}^k \triangleq \begin{pmatrix} \mathbf{z}^k \\ \boldsymbol{\beta}^k \end{pmatrix}, \quad \mathbf{u}^* \triangleq \begin{pmatrix} \mathbf{z}^* \\ \boldsymbol{\beta}^* \end{pmatrix}, \quad \mathbf{G} \triangleq \begin{pmatrix} c\mathbf{I}_{2nE} & 0_{2nE} \\ 0_{2nE} & \frac{1}{c}\mathbf{I}_{2nE} \end{pmatrix}, \quad (8)$$

if the local objective functions satisfy Assumption 1, and the dual variable  $\boldsymbol{\beta}$  is initialized such that  $\boldsymbol{\beta}^0$  lies in the column space of  $\mathbf{M}_-^T$ , then for any  $\mu > 1$ ,  $\mathbf{u}^k$  is Q-linearly convergent to its optimal point  $\mathbf{u}^*$  with respect to the  $G$ -norm

$$\|\mathbf{u}^{k+1} - \mathbf{u}^*\|_G^2 \leq \rho \|\mathbf{u}^k - \mathbf{u}^*\|_G^2 \quad (9)$$

where  $\rho$  is the convergence rate of the algorithm and

$$\rho = \frac{1}{1 + \delta}, \quad (10)$$

where

$$\delta = \min \left\{ \frac{(\mu - 1)\tilde{\sigma}_{\min}^2(\mathbf{M}_-)}{\mu\sigma_{\max}^2(\mathbf{M}_+)}, \frac{m_f}{\frac{c}{4}\sigma_{\max}^2(\mathbf{M}_+) + \frac{\mu}{c}M_f^2\tilde{\sigma}_{\min}^{-2}(\mathbf{M}_-)} \right\} \quad (11)$$

where  $\sigma_{\max}(\mathbf{M}_+)$  and  $\tilde{\sigma}_{\min}(\mathbf{M}_-)$  are the largest and smallest non-zero singular values of  $\mathbf{M}_+$  and  $\mathbf{M}_-$ , respectively. Further  $\mathbf{x}^k$  is R-linearly convergent to  $\mathbf{x}^*$ , which follows from

$$\|\mathbf{x}^{k+1} - \mathbf{x}^*\|_2^2 \leq \frac{1}{m_f} \|\mathbf{u}^k - \mathbf{u}^*\|_G^2. \quad (12)$$

*Proof:* See [24]. ■

As shown in [24],  $\mu$  and  $c$  parameters which maximize  $\delta$  are

$$\mu^* = \left( 1 + \frac{K_G^2}{2K_f^2} - \frac{K_G}{2K_f} \sqrt{\frac{K_G^2}{K_f^2} + 4} \right)^{-1}, \quad (13)$$

**Algorithm 2** : Distributed ADMM-based algorithm for consensus optimization problem in presence of node error

Input functions  $f_i$ ; Initialization: for all  $i \in \mathcal{V}$ , set  $\hat{\mathbf{x}}_i^0 = \hat{\boldsymbol{\alpha}}_i^0 = \mathbf{0}_{n \times 1} \in \mathbb{R}^n$ ; Set algorithm parameter  $c > 0$ ;  $k = 0$ ;

**for all**  $k = 0, 1, \dots$ , every node  $i$  **do**

1. Update  $\mathbf{x}_i^{k+1}$  by solving  $\nabla f_i(\mathbf{x}_i^{k+1}) + \hat{\boldsymbol{\alpha}}_i^k + 2c|\mathcal{N}_i|\mathbf{x}_i^{k+1} - c(|\mathcal{N}_i|\hat{\mathbf{x}}_i^k + \sum_{j \in \mathcal{N}_i} \hat{\mathbf{x}}_j^k) = \mathbf{0}$  ;
2. Node  $i$  sends  $\hat{\mathbf{x}}_i^{k+1} = \mathbf{x}_i^{k+1} + \mathbf{e}_{x,i}^{k+1}$  to its neighbors;
3. Update  $\hat{\boldsymbol{\alpha}}_i^{k+1} = \hat{\boldsymbol{\alpha}}_i^k + c(|\mathcal{N}_i|\hat{\mathbf{x}}_i^{k+1} - \sum_{j \in \mathcal{N}_i} \hat{\mathbf{x}}_j^{k+1})$ ;

$$c^* = \frac{2\sqrt{\mu^*}M_f}{\sigma_{\max}(\mathbf{M}_+) \tilde{\sigma}_{\min}(\mathbf{M}_-)}, \quad (14)$$

where

$$K_G = \frac{\sigma_{\max}(\mathbf{M}_+)}{\tilde{\sigma}_{\min}(\mathbf{M}_-)}, K_f = \frac{M_f}{m_f},$$

and the corresponding  $\delta$  is

$$\delta^* = \frac{1}{2K_f} \sqrt{\frac{1}{K_f^2} + \frac{4}{K_G^2}} - \frac{1}{2K_f^2}. \quad (15)$$

Note that since  $\rho = \frac{1}{1+\delta}$  is the upper bound of the convergence rate,  $c^*$  does not necessarily maximize the algorithm convergence rate, but as it is shown in [24, Figure 3], this can accelerate the algorithm.

As described and according to Algorithm 1, the nodes performing local processing in each iteration send their last estimation of the optimal point to their neighbors. In practical situations, local processing suffer from different sources of errors, such as truncation and quantization, which can potentially change the convergence behavior of the algorithm. In the next section, we analyze the performance of Algorithm 1 in presence of an additive random error at the nodes.

### III. DISTRIBUTED ADMM ALGORITHM IN PRESENCE OF NODE ERROR

In this section, we consider the scenario where the neighboring nodes only exchange a noisy version of their messages in Step 2 of Algorithm 1. We model the node error  $\mathbf{e}_{x,i}^k$  at node  $i$  in iteration  $k$ , as zero mean, additive and independent, identically distributed (i.i.d) error. We have

$$\hat{\mathbf{x}}_i^k = \mathbf{x}_i^k + \mathbf{e}_{x,i}^k, \quad \hat{\mathbf{x}}^k = \mathbf{x}^k + \mathbf{e}_x^k, \quad (16)$$

where  $\hat{\mathbf{x}}_i^k$  is the message that node  $i \in \mathcal{N}$  communicates to its neighbors at iteration  $k$ ; and  $\hat{\mathbf{x}} \in \mathbb{R}^{nN}$  and  $\mathbf{e}_x^k \in \mathbb{R}^{nN}$  denote the concatenated versions of  $\hat{\mathbf{x}}_i^k$  and  $\mathbf{e}_{x,i}^k$  for all  $i$ .

The noisy ADMM algorithm can be derived by replacing  $\mathbf{x}^{k+1}$  with  $\hat{\mathbf{x}}^{k+1}$  in (4b) and (4c). We have

$$\nabla f(\mathbf{x}^{k+1}) + \mathbf{A}^T \hat{\boldsymbol{\lambda}}^k + c\mathbf{A}^T(\mathbf{A}\mathbf{x}^{k+1} + \mathbf{B}\hat{\mathbf{z}}^k) = \mathbf{0}, \quad (17a)$$

$$\mathbf{B}^T \hat{\boldsymbol{\lambda}}^k + c\mathbf{B}^T(\mathbf{A}\hat{\mathbf{x}}^{k+1} + \mathbf{B}\hat{\mathbf{z}}^{k+1}) = \mathbf{0}, \quad (17b)$$

$$\hat{\boldsymbol{\lambda}}^{k+1} - \hat{\boldsymbol{\lambda}}^k - c(\mathbf{A}\hat{\mathbf{x}}^{k+1} + \mathbf{B}\hat{\mathbf{z}}^{k+1}) = \mathbf{0}. \quad (17c)$$

Note that  $\hat{\mathbf{x}}^{k+1}$ ,  $\hat{\mathbf{z}}^{k+1}$  and  $\hat{\boldsymbol{\lambda}}^{k+1}$  represent the variables affected by node noise  $\mathbf{e}_x^{k+1}$ . Considering  $\hat{\boldsymbol{\lambda}} = [\hat{\boldsymbol{\beta}}; \hat{\boldsymbol{\nu}}]$  with  $\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\nu}} \in \mathbb{R}^{2nE}$ , and letting  $\mathbf{z}^{k+1} = \frac{1}{2}\mathbf{M}_+^T \mathbf{x}^{k+1}$  and  $\boldsymbol{\beta}^{k+1} = \hat{\boldsymbol{\beta}}^k + \frac{c}{2}\mathbf{M}_-^T \mathbf{x}^{k+1}$ , with some mathematical

manipulations similar to those done in the derivation of (5), we have

$$\nabla f(\mathbf{x}^{k+1}) + \mathbf{M}_- \boldsymbol{\beta}^{k+1} - c\mathbf{M}_+(\hat{\mathbf{z}}^k - \mathbf{z}^{k+1}) = 0, \quad (18a)$$

$$\boldsymbol{\beta}^{k+1} = \hat{\boldsymbol{\beta}}^k + \frac{c}{2} \mathbf{M}_-^T \mathbf{x}^{k+1}, \quad (18b)$$

$$\mathbf{z}^{k+1} = \frac{1}{2} \mathbf{M}_+^T \mathbf{x}^{k+1}, \quad (18c)$$

$$\hat{\boldsymbol{\beta}}^{k+1} = \boldsymbol{\beta}^{k+1} + \frac{c}{2} \mathbf{M}_-^T \mathbf{e}_x^{k+1}, \quad (18d)$$

$$\hat{\mathbf{z}}^{k+1} = \mathbf{z}^{k+1} + \frac{1}{2} \mathbf{M}_+^T \mathbf{e}_x^{k+1}, \quad (18e)$$

which can be considered as the noisy version of iteration in (5). With some algebraic manipulations, this iteration results in the distributed noisy ADMM algorithm outlined in Algorithm 2.

In the following theorem we analyze the convergence behavior of  $d^k$  where

$$d^k = \|\hat{\mathbf{u}}^k - \mathbf{u}^*\|_G^2, \quad (19)$$

and

$$\hat{\mathbf{u}}^k = \begin{pmatrix} \hat{\mathbf{z}}^k \\ \hat{\boldsymbol{\beta}}^k \end{pmatrix}, \quad (20)$$

and  $\mathbf{u}^*$  and  $\mathbf{G}$  are defined in (8). From (18) it can be seen that  $d^k$  is a random variable, whose randomness is caused by error sequence  $\{\mathbf{e}_x^1, \dots, \mathbf{e}_x^k\}$ . In Theorem 2, we obtain an upper bound on  $\lim_{k \rightarrow \infty} \mathbb{E}[d^k]$ , which shows that  $\lim_{k \rightarrow \infty} d^k$  is bounded. As we shall demonstrate in proof of this theorem, boundedness of  $\lim_{k \rightarrow \infty} d^k$  dictates that  $\lim_{k \rightarrow \infty} \|\mathbf{x}^k - \mathbf{x}^*\|_2^2$  is finite. We provide an upper bound on the mean squared steady state error  $\lim_{k \rightarrow \infty} \mathbb{E}[\|\mathbf{x}^{k+1} - \mathbf{x}^*\|_2^2]$  of the algorithm. The proof of the theorem requires the KKT conditions for (3) to hold, i.e.,

$$\nabla f(\mathbf{x}^*) + \mathbf{M}_- \boldsymbol{\beta}^* = 0, \quad (21a)$$

$$\mathbf{M}_-^T \mathbf{x}^* = 0, \quad (21b)$$

$$\frac{1}{2} \mathbf{M}_+^T \mathbf{x}^* - \mathbf{z}^* = 0. \quad (21c)$$

**Theorem 2.** Consider the optimization problem in (3) and the iterative solution in (18) for the optimal points  $\mathbf{x}^*$  and  $\mathbf{z}^*$ . The elements of noise vector  $\mathbf{e}_x^{k+1}$  are zero mean i.i.d random variables whose variance are  $\sigma_n^2$ . Assume that local objective functions satisfy Assumption 1, and the initial value  $\boldsymbol{\beta}^0$  lies in the column space of  $\mathbf{M}_-^T$ . We have

$$\lim_{k \rightarrow \infty} \mathbb{E}_{\mathcal{F}^k} [\|\mathbf{x}^k - \mathbf{x}^*\|_2^2] \leq \frac{(4 + 3\delta)}{\delta(m_f + 2c\lambda_{\min}(\mathbf{W}))} 2cnE\sigma_n^2, \quad (22)$$

where  $\mathcal{F}^k = \{\mathbf{e}_x^1, \dots, \mathbf{e}_x^k\}$  is the history of observed error vectors up to iteration  $k$ , and  $\lambda_{\min}(\mathbf{W})$  represents the minimum eigenvalue of  $\mathbf{W}$  defined in (7).

*Proof:* See Appendix A. ■

*Remark 1.* In the noiseless scenario, we have  $\sigma_n^2 = 0$ , and Theorem 2 indicates that the upper bound on mean squared steady state error is zero. This implies that  $\mathbf{x}^*$  converges to the optimal point of problem



(3).

*Remark 2.* The upper bound on the mean squared steady state error in (22) depends on the network topology through  $\delta$ ,  $E$  and the minimum eigenvalue of matrix  $\mathbf{W}$ . There is no explicit mathematical relationship between the network topology and these parameters. We investigate this through some numerical experiments in Section IV.

Theorem 2 provides an upper bound on the mean squared steady state error of the estimate produced by Algorithm 2 under Assumption 1. We obtain a lower bound on the mean squared steady state error in Theorem 3.

**Theorem 3.** Consider the optimization problem (3) and iterative solution (18) for optimal points  $\mathbf{x}^*$  and  $\mathbf{z}^*$ . The elements of noise vector  $\mathbf{e}_x^{k+1}$  are zero mean i.i.d random variables whose variance are  $\sigma_n^2$ . Assume that the local objective functions satisfy Assumption 1. We have

$$\mathbb{E}_{\mathcal{F}^{k+1}} [\|\mathbf{x}^{k+1} - \mathbf{x}^*\|_2^2] \geq \frac{8nEc^2\sigma_n^2}{(M_f + 2c\sigma_{\max}(\mathbf{W}))^2}. \quad (23)$$

*Proof:* See Appendix B. ■

Theorem 2 provides an upper bound on the mean squared steady state error of the estimate produced by Algorithm 2 under Assumption 1. In the following theorem, we will show that the steady state error of the algorithm is bounded under a milder condition. To enable the proof we need Proposition 1 derived from the ADMM proof of convergence provided in [10], Proposition 2 and Corollary 1.

**Proposition 1.** Consider the optimization problem in the form of problem in (3), and iteration in (4) to solve it, for the general  $f(\mathbf{x})$  and  $g(\mathbf{z})$  functions, and  $\mathbf{A}$  and  $\mathbf{B}$  matrices. Assume that  $f(\mathbf{x})$  and  $g(\mathbf{z})$  are convex, proper and closed functions, and the corresponding Lagrangian function has a saddle point  $(\mathbf{x}^*, \mathbf{z}^*, \boldsymbol{\lambda}^*)$ . Let

$$V^k \triangleq \frac{1}{c} \|\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^*\|_2^2 + c \|\mathbf{B}(\mathbf{z}^k - \mathbf{z}^*)\|_2^2. \quad (24)$$

We have

$$V^{k+1} \leq V^k - c \|\mathbf{A}\mathbf{x}^{k+1} + \mathbf{B}\mathbf{z}^{k+1}\|_2^2 - c \|\mathbf{B}(\mathbf{z}^{k+1} - \mathbf{z}^*)\|_2^2, \quad (25)$$

and

$$\lim_{k \rightarrow \infty} \mathbf{A}\mathbf{x}^k + \mathbf{B}\mathbf{z}^k = \mathbf{0}, \quad \lim_{k \rightarrow \infty} \mathbf{B}(\mathbf{z}^{k+1} - \mathbf{z}^*) = \mathbf{0}. \quad (26)$$

*Proof:* See Appendix of [10]. ■

**Proposition 2.** Consider the optimization problem in (3), definition of matrices  $\mathbf{A}$  and  $\mathbf{B}$ , and the corresponding iterative solution in (5). If the local objective functions are convex, proper and closed, the iterative solution converges to an optimal point of the problem.

*Proof:* See Appendix C. ■

**Corollary 1.** We have

$$\begin{aligned} & \frac{1}{c} \|\boldsymbol{\beta}^{k+1} - \boldsymbol{\beta}^*\|_2^2 + c \|\mathbf{z}^{k+1} - \mathbf{z}^*\|_2^2 \\ & \leq \frac{1}{1 + \delta^{k+1}} \left( \frac{1}{c} \|\boldsymbol{\beta}^k - \boldsymbol{\beta}^*\|_2^2 + c \|\mathbf{z}^k - \mathbf{z}^*\|_2^2 \right), \end{aligned} \quad (27)$$

where  $\delta^0 = 0$  and  $\delta^{k+1} > 0, \forall k \geq 0$ .

*Proof:* See Appendix D. ■

In the following theorem we show that the steady state error of the noisy ADMM is bounded when local objective functions are closed, proper and convex, and do not necessarily meet more conditions, such as strong convexity.

**Theorem 4.** Consider the optimization problem in (3) and the iterative solution in (18) for the optimal points  $\mathbf{x}^*$  and  $\mathbf{z}^*$ . The elements of noise vector  $\mathbf{e}_x^{k+1}$  are zero mean i.i.d random variables whose variance

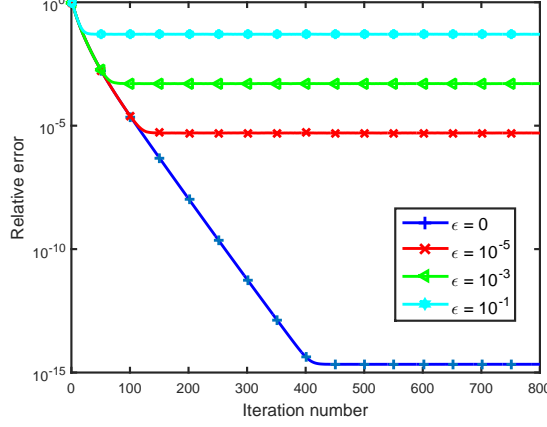


Fig. 1. Performance of Algorithm 2 when local objective functions are not strongly convex: relative error vs. iteration number over randomly generated networks for  $N = 20$ ,  $c = 1$ ,  $r = 0.5$

are  $\sigma_n^2$ , and  $\|\mathbf{e}_x^k\|_2 \leq e_{max} < \infty$ . Assume that the local objective functions are convex, closed and proper. We almost surely have

$$\lim_{k \rightarrow \infty} \|\mathbf{x}^k - \mathbf{x}^*\|_2^2 < \infty, \quad (28)$$

i.e., the steady state error of the Algorithm 2 is bounded.

*Proof:* See Appendix E ■

#### IV. NUMERICAL EVALUATION

In this section, we use Algorithm 2 to solve the following optimization problem

$$\min_{\tilde{\mathbf{x}}} \sum_{i=1}^N \frac{1}{2} \|\mathbf{y}_i - \mathbf{M}_i \tilde{\mathbf{x}}\|_2^2. \quad (29)$$

$N$  agents estimate variable  $\tilde{\mathbf{x}}$  using noisy linear observations  $\mathbf{y}_i = \mathbf{M}_i \tilde{\mathbf{x}} + \mathbf{n}_i$ , cooperatively. The variables communicated between neighbors are subject to additive node error as in (16). In these experiments, the elements of error vector are i.i.d and follow uniform distribution  $\mathcal{U}(-\epsilon, \epsilon)$ . We consider network edge density as  $r \triangleq \frac{E}{E_c}$ , where  $E_c$  is the number of edges in a corresponding complete graph. Consider  $\mathbf{x}_k^D$  and  $\mathbf{x}^C$  as respectively the distributed and centralized estimates of  $\mathbf{x}$  (at iteration  $k$ ). Our performance metrics are the relative error  $\sqrt{\mathbb{E}[\mathcal{E}_k^{DC}]}$ , where  $\mathcal{E}_k^{DC} = \frac{\|\mathbf{x}_k^D - \mathbf{x}^C\|_2^2}{\|\mathbf{x}^C\|_2^2}$ , and steady state error  $\lim_{k \rightarrow \infty} \sqrt{\mathbb{E}[\mathcal{E}_k^{DC}]}$ .

In the first set of experiments, the number of agents  $N$  is 20, and  $\tilde{\mathbf{x}} \in \mathbb{R}^{20}$  (i.e.,  $n = 20$ ) and its elements are i.i.d from  $\mathcal{N}(0, 1)$ . Measurement matrices  $\mathbf{M}_i \in \mathbb{R}^{5 \times 20}$ , which means that local objective functions are not strongly convex. The observation noise is  $\mathbf{n}_i \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_5)$  and  $\sigma^2 = 0.1$ . Figure 1 presents the relative error of the algorithm as a function of iteration index  $k$  for different values of  $\epsilon$ . In this experiment  $c = 1$  and  $r = 0.5$ . As evident the algorithm is convergent, which confirms theoretical result from Theorem 4.

In the second set of experiments the local objective functions satisfy Assumption 1, and  $\tilde{\mathbf{x}} \in \mathbb{R}^3$  (i.e.,  $n = 3$ ) and its elements are i.i.d from  $\mathcal{N}(0, 1)$ . We first generate measurement matrices  $\mathbf{M}_i \in \mathbb{R}^{3 \times 3}$ , whose elements are i.i.d from  $\mathcal{N}(0, 1)$ . Then, scale their singular values to the range  $[\sqrt{m_f}, \sqrt{M_f}]$  and rebuild  $\mathbf{M}_i$ . The observation noise is  $\mathbf{n}_i \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_3)$  and  $\sigma^2 = 0.1$ . In the following experiments, unless otherwise stated, the parameter  $c$  is set to  $c^*$  in (14).

The upper bound in Theorem 2 depends on  $\delta$  whose theoretical value is presented in (11). In numerical experiments, we can compute this parameter by estimating convergence rate of the algorithm



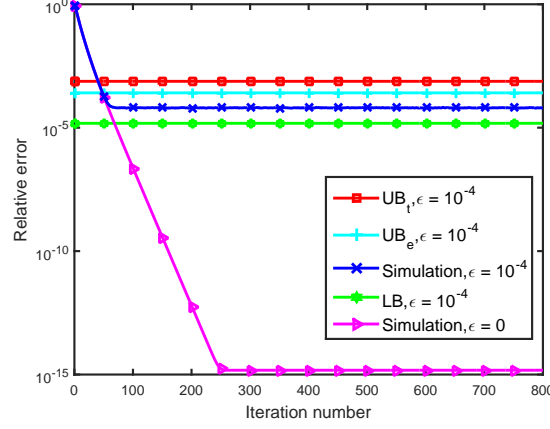


Fig. 2. Performance of Algorithm 2 when local objective functions are strongly convex and their gradients are Lipschitz continuous : relative error vs. iteration number over randomly generated networks for  $N = 20$ ,  $M_f = 10$ ,  $m_f = 1$ ,  $r = 0.5$

experimentally. Define  $\rho_e^k$  as the estimated convergence rate of the algorithm at iteration  $k$

$$\rho_e^k \triangleq \frac{\|\mathbf{u}^k - \mathbf{u}^*\|_G^2}{\|\mathbf{u}^{k-1} - \mathbf{u}^*\|_G^2}. \quad (30)$$

The geometric average rate of convergence of the algorithm is

$$\bar{\rho}_e = \left( \prod_{k=1}^K \rho_e^k \right)^{\frac{1}{K}}. \quad (31)$$

Based on (11) and (31), we have

$$\delta_e = \frac{1}{\bar{\rho}_e} - 1. \quad (32)$$

In the sequel, the upper bounds in Theorem 2 which are computed based on different  $\delta$  parameters presented in (11) and (32) are referred to as theoretical upper bound,  $UB_t$ , and experimental upper bound,  $UB_e$ , respectively. The lower bound in Theorem 3 is labeled as LB.

Figure 2 depicts the relative error of Algorithm 2 as a function of iteration number  $k$ . In this experiment, the nodes communicate over a randomly generated connected network of  $N = 20$  nodes with edge density 0.5, and  $\epsilon = 10^{-4}$ . The performance of the algorithm in error free scenario is also presented for comparison. As seen,  $UB_e$  is a tighter bound than  $UB_t$ , indicating that experimentally computed convergence rate is more accurate.

A distributed subgradient method is proposed in [31], which is convergent in presence of quantization error. Figure 3 presents the performance of this algorithm, labeled as DGD and Algorithm 2 (ADMM), for comparison. In this experiment, we have  $N = 20$  and  $r = 0.5$ . This figure shows that the ADMM-based algorithm outperforms DGD algorithm.

Figure 4 shows the effect of  $\epsilon$  on steady state error of Algorithm 2. In the current experiment set up with the least squares as the objective function of the optimization problem, one can show that indeed the steady state error of the algorithm is a linear function of the node error. This holds true not only in the performance of the algorithm (as reflected in simulations), but also in the presented lower and upper bounds.

Figure 5 presents the influence of network edge density on the steady state error of the algorithm. In this experiment networks of  $N = 20$  and  $N = 100$  nodes are generated randomly. One sees that increase in network edge density leads to little decrease in steady state error. The trends of simulation results and lower and upper bounds are consistent.

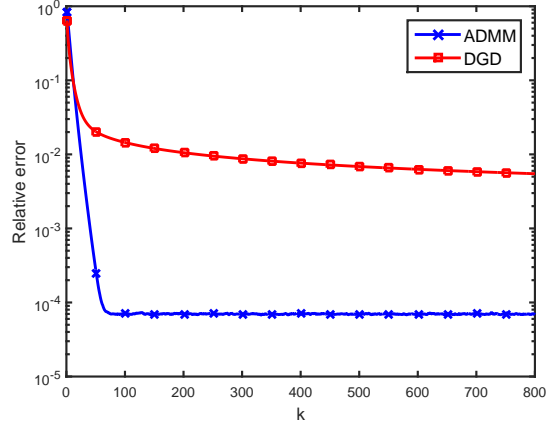


Fig. 3. Performance of Algorithm 2 (ADMM) and DGD algorithm [31], obtained with randomly generated networks for  $\epsilon = 10^{-4}$ ,  $N = 20$ ,  $M_f = 10$ ,  $m_f = 1$ , and  $r = 0.5$ .

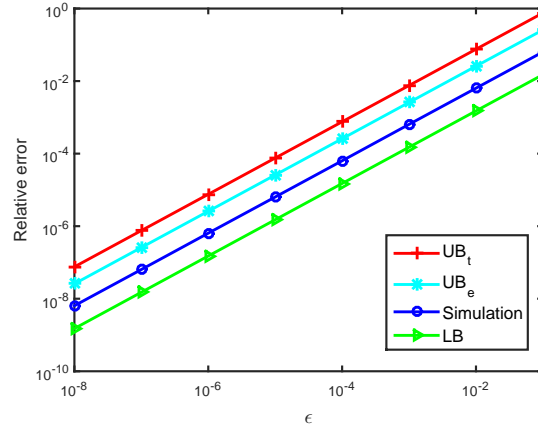


Fig. 4. Steady state error vs.  $\epsilon$ , obtained with randomly generated networks,  $N = 20$ ,  $M_f = 10$ ,  $m_f = 1$ ,  $r = 0.5$

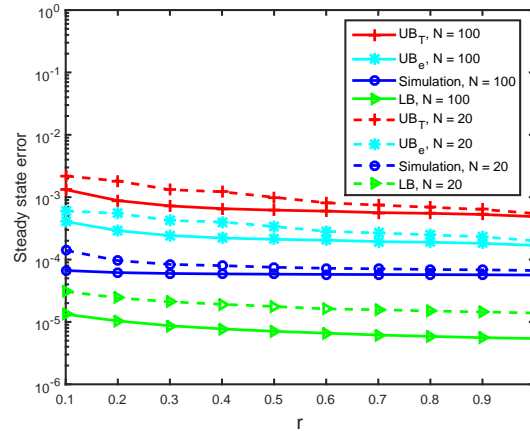


Fig. 5. Steady state error vs. edge density as a function of number of nodes  $N$ , obtained with randomly generated networks.  $\epsilon = 10^{-4}$ ,  $M_f = 10$ ,  $m_f = 1$ .

We study the performance of the noisy ADMM algorithm over regular networks as an especial case. Figure 6 depicts the effect of node degree on the steady state error of the algorithm for networks of

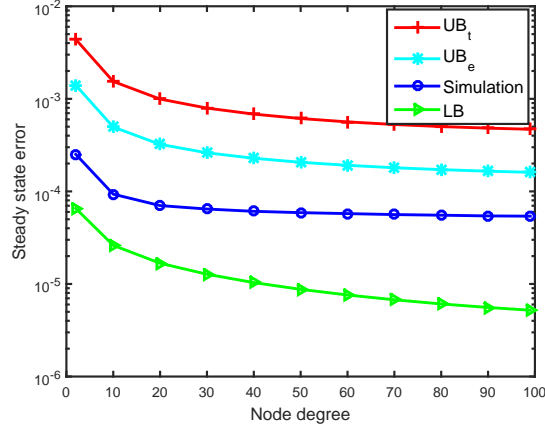


Fig. 6. Steady state error vs. node degree of regular network,  $\epsilon = 10^{-4}$ ,  $M_f = 10$ ,  $m_f = 1$ ,  $N = 100$ .

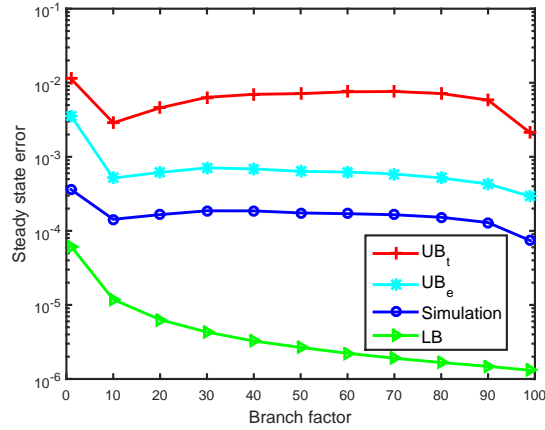


Fig. 7. Steady state error vs. branch factor of tree network,  $\epsilon = 10^{-4}$ ,  $M_f = 10$ ,  $m_f = 1$ ,  $N = 100$

$N = 100$  nodes. As evident, the slope of the curve is a decreasing function of node degree, and varying node degree in the range of  $[30 \ 99]$  does not improve steady state error significantly. The upper bounds which are functions of  $E$  and  $\delta$ , predict this behavior properly. The lower bound is just a function of  $E$ , and does not predict the slope of steady state error curve.

We consider tree network as another especial case. Figure 7 demonstrates the influence of branch factor parameter of the tree network on steady state error of the algorithm. It can be seen that experimental upper bound predicts the performance of the algorithm better than the other bounds, and proposed theoretical upper bound is an acceptable indicator of the algorithm behavior. From the last three sets of experiments, we can conclude that the upper bounds can predict effects of network topology on the steady state error more accurately.

Another parameter which affects the performance of the algorithm is  $c$ . Figure 8 shows the effect of  $c$  parameter on the steady state error over the randomly generated networks. The connectivity ratio in  $N = 20$  and  $N = 100$  scenarios are 0.5 and 0.2, respectively. One sees that, a lower  $c$  results in a lower steady state error which is also predicted by the proposed lower and experimental upper bounds. The mismatch between theoretical upper bound and simulation shows that the effect of  $c$  on the convergence rate of the algorithm is not predicted properly by (10).

As discussed, the convergence rate and steady state error of the algorithm output depends on parameter  $c$ . The value of the parameter  $c$  is to be selected for minimized steady state error and maximized convergence rate. According to our observations, the performance of the noisy ADMM algorithm in early iterations is

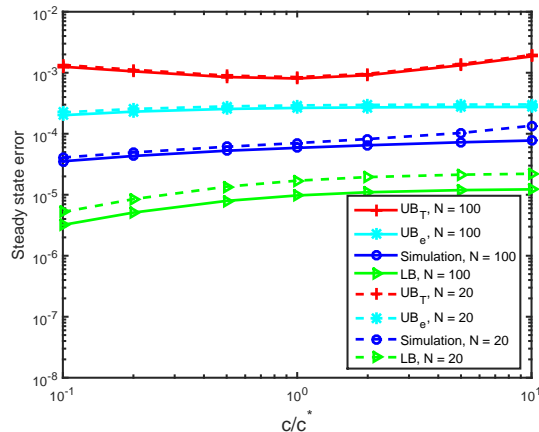


Fig. 8. Steady state error vs.  $c$  parameter as a function of number of nodes  $N$ , obtained with randomly generated networks.  $\epsilon = 10^{-4}$ ,  $M_f = 10$ ,  $m_f = 1$ .

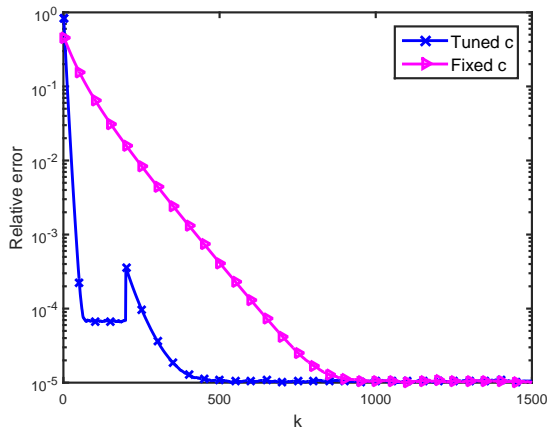


Fig. 9. Comparison of performance of tuned and fixed  $c$  algorithms,  $\epsilon = 10^{-4}$ ,  $M_f = 10$ ,  $m_f = 1$ ,  $N = 20$ ,  $r = 0.5$

similar to the noiseless case. A smaller value of  $c$  results in a smaller steady state error, but it does not necessarily lead to a high convergence rate. A reasonable tuning policy is to set parameter  $c$  to accelerate the convergence in early iterations, and then to reduce the parameter to aim for a small steady state error. Figure 9 depicts the performance of this tuning method. In this experiment, parameter  $c$  is set to  $c^*$  provided in (14) in the first 200 iterations, and then it is reduced by a factor of 0.01. The performance of the algorithm with fixed parameter  $c$  set to  $0.01c^*$  is also presented for comparison. The results confirm faster convergence rate and improved steady state error when the proposed tuning policy is in effect.

We also examined the effects of  $M_f$ , Lipschitz constant, and  $m_f$ , strong convexity module, on the steady state error of the algorithm (not reported here). We observed that the influence of these parameters is accurately predicted by the proposed lower and experimental upper bounds. The theoretical upper bound, however, does not reflect the effects of these parameters properly.

## V. CONCLUSIONS

We analyzed the effect of additive node error on the performance of distributed ADMM algorithm for consensus optimization over a connected network. Analytical upper and lower bounds were provided on the mean squared steady state error of the algorithm, in the case that local objective functions are strongly convex and have Lipschitz continuous gradients. The analysis quantifies how the provided bounds depend on different factors such as noise variance and network topology. To accelerate the algorithm and also

reduce the steady state error, a method was proposed to tune the algorithm parameter. In addition, it was shown that if the local objective functions are proper, closed and convex, for a bounded and random node error, the steady state error of the noisy ADMM algorithm for consensus is bounded. Numerical results validated the theoretical analyses and demonstrated the role of different system and network parameters. Analysis of the algorithm convergence behavior over the networks whose links suffer from additive noise can be considered as an important next research steps.

## APPENDIX A PROOF OF THEOREM 2

We first derive an upper bound on  $\lim_{k \rightarrow \infty} \mathbb{E}[d^k]$ . We have

$$\mathbb{E}_{\mathcal{F}^{k+1}}[d^{k+1}] = \mathbb{E}_{\mathcal{F}^k} \{ \mathbb{E}_{e_x^{k+1}}[d^{k+1} | \mathcal{F}^k] \}, \quad (33)$$

where

$$\begin{aligned} \mathbb{E}_{e_x^{k+1}}[d^{k+1} | \mathcal{F}^k] &= \mathbb{E}_{e_x^{k+1}} \left[ \frac{1}{c} \|\hat{\beta}^{k+1} - \beta^*\|_2^2 + c \|\hat{\mathbf{z}}^{k+1} - \mathbf{z}^*\|_2^2 | \mathcal{F}^k \right] \\ &\stackrel{(a)}{=} \mathbb{E}_{e_x^{k+1}} \left[ \frac{1}{c} \|\beta^{k+1} - \beta^* + \frac{c}{2} \mathbf{M}_-^T \mathbf{e}_x^{k+1}\|_2^2 | \mathcal{F}^k \right] \\ &\quad + \mathbb{E}_{e_x^{k+1}} \left[ c \|\mathbf{z}^{k+1} - \mathbf{z}^* + \frac{1}{2} \mathbf{M}_+^T \mathbf{e}_x^{k+1}\|_2^2 | \mathcal{F}^k \right] \\ &\stackrel{(b)}{=} \frac{1}{c} \|\beta^{k+1} - \beta^*\|_2^2 + c \|\mathbf{z}^{k+1} - \mathbf{z}^*\|_2^2 + c \mathbb{E}[(\mathbf{e}_x^{k+1})^T \mathbf{W} \mathbf{e}_x^{k+1}] \\ &= \frac{1}{c} \|\beta^{k+1} - \beta^*\|_2^2 + c \|\mathbf{z}^{k+1} - \mathbf{z}^*\|_2^2 + 2cnE\sigma_n^2 \\ &\stackrel{(c)}{\leq} \frac{1}{1+\delta} \left\{ \frac{1}{c} \|\hat{\beta}^k - \beta^*\|_2^2 + c \|\hat{\mathbf{z}}^k - \mathbf{z}^*\|_2^2 \right\} + 2cnE\sigma_n^2 \\ &= \frac{1}{1+\delta} d^k + 2cnE\sigma_n^2, \end{aligned} \quad (34)$$

where (a) follows from (18d) and (18e), (b) follows from the fact that the elements of noise vector  $\mathbf{e}_x^{k+1}$  are zero mean i.i.d random variables and (c) follows from (9) and (18a)- (18c). From (33) and (34), we have

$$\mathbb{E}_{\mathcal{F}^{k+1}}[d^{k+1}] \leq \frac{1}{1+\delta} \mathbb{E}_{\mathcal{F}^k}[d^k] + 2cnE\sigma_n^2. \quad (35)$$

Recursively using (35), we have

$$\mathbb{E}_{\mathcal{F}^{k+1}}[d^{k+1}] \leq \frac{d^0}{(1+\delta)^{k+1}} + \sum_{\ell=0}^k \frac{1}{(1+\delta)^\ell} 2cnE\sigma_n^2, \quad (36)$$

which leads to

$$\lim_{k \rightarrow \infty} \mathbb{E}_{\mathcal{F}^k}[d^k] \leq \frac{1+\delta}{\delta} 2cnE\sigma_n^2. \quad (37)$$

To prove (22), we use the update equation for  $\mathbf{x}$  in (18a), and its corresponding KKT condition (21a). Our goal is to derive a mathematical relationship between  $\mathbf{x}^{k+1} - \mathbf{x}^*$ ,  $\hat{\beta}^k - \beta^*$  and  $\hat{\mathbf{z}}^k - \mathbf{z}^*$ . To this aim we combine (18a) and (21a) as

$$\nabla f(\mathbf{x}^{k+1}) - \nabla f(\mathbf{x}^*) + \mathbf{M}_-(\beta^{k+1} - \beta^*) - c\mathbf{M}_+(\hat{\mathbf{z}}^k - \mathbf{z}^{k+1}) = 0. \quad (38)$$

Replacing (18b) and (18c) in (38), we have

$$\begin{aligned} \nabla f(\mathbf{x}^{k+1}) - \nabla f(\mathbf{x}^*) + \frac{c}{2} \mathbf{M}_- \mathbf{M}_-^T \mathbf{x}^{k+1} + \frac{c}{2} \mathbf{M}_+ \mathbf{M}_+^T \mathbf{x}^{k+1} \\ = -\mathbf{M}_- (\hat{\boldsymbol{\beta}}^k - \boldsymbol{\beta}^*) + c \mathbf{M}_+ \hat{\mathbf{z}}^k. \end{aligned} \quad (39)$$

According to (7), combining (39) with (21b) and (21c) leads to

$$\begin{aligned} \nabla f(\mathbf{x}^{k+1}) - \nabla f(\mathbf{x}^*) + 2c \mathbf{W}(\mathbf{x}^{k+1} - \mathbf{x}^*) \\ = -\mathbf{M}_- (\hat{\boldsymbol{\beta}}^k - \boldsymbol{\beta}^*) + c \mathbf{M}_+ (\hat{\mathbf{z}}^k - \mathbf{z}^*). \end{aligned} \quad (40)$$

The inner product of  $\mathbf{x}^{k+1} - \mathbf{x}^*$  and (40) is

$$\begin{aligned} \langle \mathbf{x}^{k+1} - \mathbf{x}^*, \nabla f(\mathbf{x}^{k+1}) - \nabla f(\mathbf{x}^*) \rangle \\ + 2c \langle \mathbf{x}^{k+1} - \mathbf{x}^*, \mathbf{W}(\mathbf{x}^{k+1} - \mathbf{x}^*) \rangle \\ = -\langle \mathbf{M}_-^T (\mathbf{x}^{k+1} - \mathbf{x}^*), \hat{\boldsymbol{\beta}}^k - \boldsymbol{\beta}^* \rangle \\ + c \langle \mathbf{M}_+^T (\mathbf{x}^{k+1} - \mathbf{x}^*), \hat{\mathbf{z}}^k - \mathbf{z}^* \rangle \\ \stackrel{(a)}{=} -\frac{2}{c} \langle \hat{\boldsymbol{\beta}}^k - \boldsymbol{\beta}^*, (\boldsymbol{\beta}^{k+1} - \boldsymbol{\beta}^*) - (\hat{\boldsymbol{\beta}}^k - \boldsymbol{\beta}^*) \rangle \\ + 2c \langle \hat{\mathbf{z}}^k - \mathbf{z}^*, \mathbf{z}^{k+1} - \mathbf{z}^* \rangle, \end{aligned} \quad (41)$$

where (a) follows from (18b) and (18c). Since  $\mathbf{W}$  is a positive definite matrix, and based on strong convexity of  $f(\mathbf{x})$ , the lower bound on the LHS of (41) is

$$(m_f + 2c\lambda_{\min}(\mathbf{W})) \|\mathbf{x}^{k+1} - \mathbf{x}^*\|_2^2. \quad (42)$$

Replacing (42) in (41) leads to

$$\begin{aligned} (m_f + 2c\lambda_{\min}(\mathbf{W})) \|\mathbf{x}^{k+1} - \mathbf{x}^*\|_2^2 \\ \leq -\frac{2}{c} \langle \hat{\boldsymbol{\beta}}^k - \boldsymbol{\beta}^*, (\boldsymbol{\beta}^{k+1} - \boldsymbol{\beta}^*) - (\hat{\boldsymbol{\beta}}^k - \boldsymbol{\beta}^*) \rangle \\ + 2c \langle \hat{\mathbf{z}}^k - \mathbf{z}^*, \mathbf{z}^{k+1} - \mathbf{z}^* \rangle \\ \leq \frac{1}{c} \|\boldsymbol{\beta}^{k+1} - \boldsymbol{\beta}^*\|_2^2 + \frac{3}{c} \|\hat{\boldsymbol{\beta}}^k - \boldsymbol{\beta}^*\|_2^2 + c \|\hat{\mathbf{z}}^k - \mathbf{z}^*\|_2^2 \\ + c \|\mathbf{z}^{k+1} - \mathbf{z}^*\|_2^2 \\ \stackrel{(a)}{\leq} \left(3 + \frac{1}{1+\delta}\right) \|\hat{\mathbf{u}}^k - \mathbf{u}^*\|_G^2 \\ \stackrel{(b)}{=} \left(3 + \frac{1}{1+\delta}\right) d^k, \end{aligned} \quad (43)$$

where (a) and (b) follow from (8) and (9), and (19), respectively. Combining (43) and (37), we have

$$\lim_{k \rightarrow \infty} \mathbb{E}_{\mathcal{F}^{k+1}} [\|\mathbf{x}^{k+1} - \mathbf{x}^*\|_2^2] \leq \frac{4 + 3\delta}{\delta(m_f + 2c\lambda_{\min}(\mathbf{W}))} 2cnE\sigma_n^2, \quad (44)$$

which completes the proof.

## APPENDIX B PROOF OF THEOREM 3

Subtracting (21a) from (18a), we have

$$\nabla f(\mathbf{x}^{k+1}) - \nabla f(\mathbf{x}^*) + \mathbf{M}_- (\boldsymbol{\beta}^{k+1} - \boldsymbol{\beta}^*) - c \mathbf{M}_+ (\hat{\mathbf{z}}^k - \mathbf{z}^{k+1}) = 0. \quad (45)$$



Replacing (18b) and (18c) in (45) yields

$$\begin{aligned} & \nabla f(\mathbf{x}^{k+1}) - \nabla f(\mathbf{x}^*) + 2c\mathbf{W}(\mathbf{x}^{k+1} - \mathbf{x}^*) \\ &= c\mathbf{M}_+(\hat{\mathbf{z}}^k - \mathbf{z}^*) - \mathbf{M}_-(\hat{\boldsymbol{\beta}}^k - \boldsymbol{\beta}^*). \end{aligned} \quad (46)$$

Writing  $\hat{\mathbf{z}}^k$  and  $\hat{\boldsymbol{\beta}}^k$  in terms of  $\mathbf{z}^k$ ,  $\boldsymbol{\beta}^k$  and  $\mathbf{e}_x^k$  in (46) leads to

$$\begin{aligned} & \|\nabla f(\mathbf{x}^{k+1}) - \nabla f(\mathbf{x}^*) + 2c\mathbf{W}(\mathbf{x}^{k+1} - \mathbf{x}^*)\|_2^2 \\ &= \|c\mathbf{M}_+(\mathbf{z}^k - \mathbf{z}^*) - \mathbf{M}_-(\boldsymbol{\beta}^k - \boldsymbol{\beta}^*) \\ & \quad + \frac{c}{2}(\mathbf{M}_+\mathbf{M}_+^T - \mathbf{M}_-\mathbf{M}_-^T)\mathbf{e}_x^k\|_2^2. \end{aligned} \quad (47)$$

The upper bound on the square root of the LHS of (47) is

$$\begin{aligned} & \|\nabla f(\mathbf{x}^{k+1}) - \nabla f(\mathbf{x}^*) + 2c\mathbf{W}(\mathbf{x}^{k+1} - \mathbf{x}^*)\|_2 \\ & \leq \|\nabla f(\mathbf{x}^{k+1}) - \nabla f(\mathbf{x}^*)\|_2 + \|2c\mathbf{W}(\mathbf{x}^{k+1} - \mathbf{x}^*)\|_2 \\ & \stackrel{(a)}{\leq} (M_f + 2c\sigma_{\max}(\mathbf{W}))\|\mathbf{x}^{k+1} - \mathbf{x}^*\|_2, \end{aligned} \quad (48)$$

where (a) follows from Lipschitz continuity property of  $\nabla f(\mathbf{x})$ . The RHS of (47) can be rewritten as follows

$$\begin{aligned} & \|c\mathbf{M}_+(\mathbf{z}^k - \mathbf{z}^*) - \mathbf{M}_-(\boldsymbol{\beta}^k - \boldsymbol{\beta}^*) + \frac{c}{2}(\mathbf{M}_+\mathbf{M}_+^T - \mathbf{M}_-\mathbf{M}_-^T)\mathbf{e}_x^k\|_2^2 \\ &= 2\langle c\mathbf{M}_+(\mathbf{z}^k - \mathbf{z}^*) - \mathbf{M}_-(\boldsymbol{\beta}^k - \boldsymbol{\beta}^*), \frac{c}{2}(\mathbf{M}_+\mathbf{M}_+^T - \mathbf{M}_-\mathbf{M}_-^T)\mathbf{e}_x^k \rangle \\ & \quad + \|c\mathbf{M}_+(\mathbf{z}^k - \mathbf{z}^*) - \mathbf{M}_-(\boldsymbol{\beta}^k - \boldsymbol{\beta}^*)\|_2^2 \\ & \quad + \|\frac{c}{2}(\mathbf{M}_+\mathbf{M}_+^T - \mathbf{M}_-\mathbf{M}_-^T)\mathbf{e}_x^k\|_2^2. \end{aligned} \quad (49)$$

From (47), (48) and (49), we have

$$\begin{aligned} & (M_f + 2c\sigma_{\max}(\mathbf{W}))^2\|\mathbf{x}^{k+1} - \mathbf{x}^*\|_2^2 \\ & \geq 2\langle c\mathbf{M}_+(\mathbf{z}^k - \mathbf{z}^*) - \mathbf{M}_-(\boldsymbol{\beta}^k - \boldsymbol{\beta}^*), \frac{c}{2}(\mathbf{M}_+\mathbf{M}_+^T - \mathbf{M}_-\mathbf{M}_-^T)\mathbf{e}_x^k \rangle \\ & \quad + \|c\mathbf{M}_+(\mathbf{z}^k - \mathbf{z}^*) - \mathbf{M}_-(\boldsymbol{\beta}^k - \boldsymbol{\beta}^*)\|_2^2 \\ & \quad + \|\frac{c}{2}(\mathbf{M}_+\mathbf{M}_+^T - \mathbf{M}_-\mathbf{M}_-^T)\mathbf{e}_x^k\|_2^2. \end{aligned} \quad (50)$$

Taking average from both sides of (50), and considering that the elements of  $\mathbf{e}_x^k$  are zero mean random variables independent of  $\mathbf{z}^k$  and  $\boldsymbol{\beta}^k$ , we have

$$\begin{aligned} & \mathbb{E}_{\mathcal{F}^{k+1}}[\|\mathbf{x}^{k+1} - \mathbf{x}^*\|_2^2] \\ & \geq \frac{c^2}{(M_f + 2c\sigma_{\max}(\mathbf{W}))^2} \mathbb{E}[\|(\mathbf{L}_+ - \mathbf{L}_-)\mathbf{e}_x^k\|_2^2], \end{aligned} \quad (51)$$

where  $\mathbf{L}_+ = \frac{1}{2}\mathbf{M}_+\mathbf{M}_+^T$  and  $\mathbf{L}_- = \frac{1}{2}\mathbf{M}_-\mathbf{M}_-^T$  are, respectively, the extended signless and signed Laplacian matrices of the underlying network. According to the structure of these matrices, the  $(i, j)$  block of  $\mathbf{P} = \mathbf{L}_+ - \mathbf{L}_-$  is as follows

$$\mathbf{P}(i, j) = \begin{cases} \mathbf{0}_n & i = j, \text{ or } (i, j) \notin \mathcal{A} \\ 2\mathbf{I}_n & (i, j) \in \mathcal{A} \end{cases}. \quad (52)$$

According to (52), following some algebraic manipulations, we have

$$\mathbb{E}_{\mathcal{F}^{k+1}}[\|\mathbf{x}^{k+1} - \mathbf{x}^*\|_2^2] \geq \frac{8nEc^2\sigma_n^2}{(M_f + 2c\sigma_{\max}(\mathbf{W}))^2}, \quad (53)$$

which completes the proof.

### APPENDIX C PROOF OF PROPOSITION 2

According to the definition of matrices  $\mathbf{A}$  and  $\mathbf{B}$ , the feasible set of problem in (3) is nonempty. This fact and the convexity of local objective functions guarantee that the Lagrangian function corresponding to (3) has a saddle point. If local objective functions are also proper and closed, inequality (25) in Proposition 1 is valid for this problem. From (4c), we have

$$\mathbf{A}\mathbf{x}^{k+1} + \mathbf{B}\mathbf{z}^{k+1} = \frac{1}{c}(\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k). \quad (54)$$

Since  $\boldsymbol{\lambda}^k = [\boldsymbol{\beta}^k; -\boldsymbol{\beta}^k]$ , and  $\mathbf{B} = [-\mathbf{I}_{2nE}; -\mathbf{I}_{2nE}]$ , and based on (54), (25) can be rewritten as follows

$$\begin{aligned} & \frac{1}{c} \|\boldsymbol{\beta}^{k+1} - \boldsymbol{\beta}^*\|_2^2 + c \|\mathbf{z}^{k+1} - \mathbf{z}^*\|_2^2 \\ & \leq \frac{1}{c} \|\boldsymbol{\beta}^k - \boldsymbol{\beta}^*\|_2^2 + c \|\mathbf{z}^k - \mathbf{z}^*\|_2^2 \\ & \quad - \frac{1}{c} \|\boldsymbol{\beta}^{k+1} - \boldsymbol{\beta}^k\|_2^2 - c \|\mathbf{z}^{k+1} - \mathbf{z}^k\|_2^2, \end{aligned} \quad (55)$$

and (26) is equivalent to

$$\lim_{k \rightarrow \infty} \boldsymbol{\beta}^{k+1} - \boldsymbol{\beta}^k = \mathbf{0}, \quad \lim_{k \rightarrow \infty} \mathbf{z}^{k+1} - \mathbf{z}^k = \mathbf{0}. \quad (56)$$

Replacing (56) in (5), we have

$$\lim_{k \rightarrow \infty} \nabla f(\mathbf{x}^k) + \mathbf{M}_- \boldsymbol{\beta}^k = \mathbf{0}, \quad (57a)$$

$$\lim_{k \rightarrow \infty} \mathbf{M}_-^T \mathbf{x}^k = \mathbf{0}, \quad (57b)$$

$$\lim_{k \rightarrow \infty} \frac{1}{2} \mathbf{M}_+^T \mathbf{x}^k - \mathbf{z}^k = \mathbf{0}. \quad (57c)$$

Comparing (57) and (21), it can be seen that as  $k \rightarrow \infty$ , primal and dual variables satisfy KKT conditions, and hence the iteration converges to an optimal point. In the other words, (56) is a sufficient condition for attaining an optimal point, which completes the proof.

### APPENDIX D PROOF OF COROLLARY 1

From (55) in Proposition 2, we have

$$\begin{aligned} 0 & \leq \frac{1}{c} \|\boldsymbol{\beta}^{k+1} - \boldsymbol{\beta}^k\|_2^2 + c \|\mathbf{z}^{k+1} - \mathbf{z}^k\|_2^2 \\ & \leq \frac{1}{c} \|\boldsymbol{\beta}^k - \boldsymbol{\beta}^*\|_2^2 + c \|\mathbf{z}^k - \mathbf{z}^*\|_2^2, \end{aligned} \quad (58)$$

which is equivalent to

$$\begin{aligned} & \frac{1}{c} \|\boldsymbol{\beta}^{k+1} - \boldsymbol{\beta}^k\|_2^2 + c \|\mathbf{z}^{k+1} - \mathbf{z}^k\|_2^2 \\ & = \eta^{k+1} \left( \frac{1}{c} \|\boldsymbol{\beta}^k - \boldsymbol{\beta}^*\|_2^2 + c \|\mathbf{z}^k - \mathbf{z}^*\|_2^2 \right), \end{aligned} \quad (59)$$

where  $0 < \eta^{k+1} \leq 1, \forall k \geq 0$ . Note that, based on Proposition 2, the LHS of (59) being zero, guarantees that its RHS is also zero, and hence we can consider  $0 < \eta^{k+1}$  in all cases. From (55) and (59), we have

$$\begin{aligned} & \frac{1}{c} \|\beta^{k+1} - \beta^*\|_2^2 + c \|\mathbf{z}^{k+1} - \mathbf{z}^*\|_2^2 \\ & \leq \frac{1}{1 + \delta^{k+1}} \left( \frac{1}{c} \|\beta^k - \beta^*\|_2^2 + c \|\mathbf{z}^k - \mathbf{z}^*\|_2^2 \right), \end{aligned} \quad (60)$$

where  $1/(1 + \delta^{k+1}) = 1 - \eta^{k+1}$ ,  $\delta^0 = 0$  and  $\delta^{k+1} > 0, \forall k \geq 0$ , which completes the proof.

## APPENDIX E PROOF OF THEOREM 4

Similar to proof of Theorem 2, we have

$$\begin{aligned} \mathbb{E}_{\mathcal{F}^{k+1}}[d^{k+1}] &= \mathbb{E}_{\mathcal{F}^k} \{ \mathbb{E}_{e^{k+1}}[d^{k+1} | \mathcal{F}^k] \} \\ &\stackrel{(a)}{\leq} \frac{1}{1 + \delta^{k+1}} \mathbb{E}_{\mathcal{F}^k}[d^k] + 2cnE\sigma_n^2, \end{aligned} \quad (61)$$

where (a) follows from (27) in Corollary 1. By recursive use of (61), we have

$$\mathbb{E}_{\mathcal{F}^{k+1}}[d^{k+1}] \leq \frac{d^0}{\prod_{\ell=1}^{k+1} (1 + \delta^\ell)} + \sum_{\ell=0}^k \frac{1}{\prod_{i=0}^{\ell} (1 + \delta^i)} 2cnE\sigma_n^2. \quad (62)$$

Since  $\delta^k > 0, \forall k > 0$ , we have

$$\lim_{k \rightarrow \infty} \mathbb{E}_{\mathcal{F}^k}[d^k] < \infty, \quad (63)$$

and hence we almost surely have,

$$\lim_{k \rightarrow \infty} d^k < \infty, \quad (64)$$

which implies

$$\begin{aligned} \lim_{k \rightarrow \infty} \|\hat{\beta}^{k+1} - \beta^*\|_2 &< \infty, \\ \lim_{k \rightarrow \infty} \|\hat{\mathbf{z}}^{k+1} - \mathbf{z}^*\|_2 &< \infty. \end{aligned} \quad (65)$$

From (18d), (18e), (21b) and (21c), we have

$$\begin{aligned} \hat{\beta}^{k+1} - \beta^* &= \hat{\beta}^k - \beta^* + \frac{c}{2} \mathbf{M}_-^T (\mathbf{x}^{k+1} - \mathbf{x}^*) + \frac{c}{2} \mathbf{M}_-^T \mathbf{e}_x^{k+1}, \\ \hat{\mathbf{z}}^{k+1} - \mathbf{z}^* &= \frac{1}{2} \mathbf{M}_+^T (\mathbf{x}^{k+1} - \mathbf{x}^*) + \frac{1}{2} \mathbf{M}_+^T \mathbf{e}_x^{k+1}. \end{aligned} \quad (66)$$

Since  $\|\mathbf{e}_x^k\|_2 \leq e_{max} < \infty$ , and based on the definition of  $\mathbf{M}_-$  and  $\mathbf{M}_+$ , from (65) and (66) we have

$$\lim_{k \rightarrow \infty} \|\mathbf{x}^k - \mathbf{x}^*\|_2^2 < \infty,$$

which completes the proof.

## REFERENCES

- [1] L. Majzoobi and F. Lahouti, "Analysis of distributed admm algorithm for consensus optimization in presence of error," in *41st IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 4831–4835.
- [2] S. H. Low and D. E. Lapsley, "Optimization flow control-I: basic algorithm and convergence," *IEEE/ACM Transactions on Networking*, vol. 7, no. 6, pp. 861–874, 1999.
- [3] T. Erseghe, "Distributed optimal power flow using admm," *IEEE transactions on power systems*, vol. 29, no. 5, pp. 2370–2380, 2014.
- [4] M. Rabbat and R. Nowak, "Distributed optimization in sensor networks," in *ACM Proceedings of the 3rd international symposium on Information processing in sensor networks*, 2004, pp. 20–27.

- [5] A. Nedić and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, 2009.
- [6] B. Johansson, M. Rabi, and M. Johansson, "A randomized incremental subgradient method for distributed optimization in networked systems," *SIAM Journal on Optimization*, vol. 20, no. 3, pp. 1157–1170, 2009.
- [7] E. Ghadimi, I. Shames, and M. Johansson, "Multi-step gradient methods for networked optimization," *IEEE Transactions on Signal Processing*, vol. 61, no. 21, pp. 5417–5429, 2013.
- [8] N. Li, "Distributed optimization in power networks and general multi-agent systems," Ph.D. dissertation, California Institute of Technology, 2013.
- [9] T.-H. Chang, A. Nedic, and A. Scaglione, "Distributed constrained optimization by consensus-based primal-dual perturbation method," *IEEE Transactions on Automatic Control*, vol. 59, no. 6, pp. 1524–1538, 2014.
- [10] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.
- [11] C. Shen, T.-H. Chang, K.-Y. Wang, Z. Qiu, and C.-Y. Chi, "Distributed robust multicell coordinated beamforming with imperfect csi: An admm approach," *IEEE Transactions on signal processing*, vol. 60, no. 6, pp. 2988–3003, 2012.
- [12] F. Lin, M. Fardad, and M. R. Jovanović, "Design of optimal sparse feedback gains via the alternating direction method of multipliers," *IEEE Transactions on Automatic Control*, vol. 58, no. 9, pp. 2426–2431, 2013.
- [13] J. Liu, P. Musialski, P. Wonka, and J. Ye, "Tensor completion for estimating missing values in visual data," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 1, pp. 208–220, 2013.
- [14] P. Danaher, P. Wang, and D. M. Witten, "The joint graphical lasso for inverse covariance estimation across multiple classes," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 76, no. 2, pp. 373–397, 2014.
- [15] X. Qing, G. Hu, and X. Wang, "Robust compressed sensing with bounded and structured uncertainties," *IET Signal Processing*, vol. 8, no. 7, pp. 783–791, 2014.
- [16] W. Deng and W. Yin, "On the global and linear convergence of the generalized alternating direction method of multipliers," *Journal of Scientific Computing*, pp. 1–28, 2012.
- [17] B. He and X. Yuan, "On the  $o(1/n)$  convergence rate of the douglas-rachford alternating direction method," *SIAM Journal on Numerical Analysis*, vol. 50, no. 2, pp. 700–709, 2012.
- [18] W. Deng, M.-J. Lai, Z. Peng, and W. Yin, "Parallel multi-block admm with  $o(1/k)$  convergence," *Journal of Scientific Computing*, pp. 1–25, 2014.
- [19] T. Goldstein, B. O'Donoghue, S. Setzer, and R. Baraniuk, "Fast alternating direction optimization methods," *SIAM Journal on Imaging Sciences*, vol. 7, no. 3, pp. 1588–1623, 2014.
- [20] P. Giselsson and S. Boyd, "Linear convergence and metric selection for douglas-rachford splitting and admm," *IEEE Transactions on Automatic Control*, vol. 62, no. 2, pp. 532–544, 2017.
- [21] G. Mateos, J. A. Bazerque, and G. B. Giannakis, "Distributed sparse linear regression," *IEEE Transactions on Signal Processing*, vol. 58, no. 10, pp. 5262–5276, 2010.
- [22] R. H. Etemad and F. Lahouti, "Resilient decentralized consensus-based state estimation for smart grid in presence of false data," in *41st IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 3466–3470.
- [23] J. F. Mota, J. M. Xavier, P. M. Aguiar, and M. Puschel, "D-admm: A communication-efficient distributed algorithm for separable optimization," *IEEE Transactions on Signal Processing*, vol. 61, no. 10, pp. 2718–2723, 2013.
- [24] W. Shi, Q. Ling, K. Yuan, G. Wu, and W. Yin, "On the linear convergence of the admm in decentralized consensus optimization," *IEEE Transactions on Signal Processing*, vol. 62, no. 7, pp. 1750–1761, 2014.
- [25] M. Ma, A. N. Nikolakopoulos, and G. B. Giannakis, "Fast decentralized learning via hybrid consensus admm," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Calgary, Alberta, Canada, 2018.
- [26] A. Nedić and A. Olshevsky, "Distributed optimization over time-varying directed graphs," *IEEE Transactions on Automatic Control*, vol. 60, no. 3, pp. 601–615, 2015.
- [27] L. Majzoobi, V. Shah-Mansouri, and F. Lahouti, "Analysis of distributed admm algorithm for consensus optimisation over lossy networks," *IET Signal Processing*, vol. 12, no. 6, pp. 786–794, 2018.
- [28] T.-H. Chang, M. Hong, and X. Wang, "Multi-agent distributed optimization via inexact consensus admm," *IEEE Transactions on Signal Processing*, vol. 63, no. 2, pp. 482–497, 2015.
- [29] Q. Ling, W. Shi, G. Wu, and A. Ribeiro, "Dlm: Decentralized linearized alternating direction method of multipliers," *IEEE Transactions Signal Processing*, vol. 63, no. 15, pp. 4051–4064, 2015.
- [30] A. Mokhtari, W. Shi, Q. Ling, and A. Ribeiro, "Decentralized quadratically approximated alternating direction method of multipliers," in *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 2015, pp. 795–799.
- [31] A. Nedic, A. Olshevsky, A. Ozdaglar, and J. N. Tsitsiklis, "Distributed subgradient methods and quantization effects," in *IEEE 47th Conference on Decision and Control (CDC)*, 2008, pp. 4177–4184.
- [32] I. D. Schizas, A. Ribeiro, and G. B. Giannakis, "Consensus in ad hoc wsns with noisy links—part i: Distributed estimation of deterministic signals," *IEEE Transactions on Signal Processing*, vol. 56, no. 1, pp. 350–364, 2008.
- [33] T. Erseghe, D. Zennaro, E. Dall'Anese, and L. Vangelista, "Fast consensus by the alternating direction multipliers method," *IEEE Transactions on Signal Processing*, vol. 59, no. 11, pp. 5523–5537, 2011.
- [34] D. Marco and D. L. Neuhoff, "The validity of the additive noise model for uniform scalar quantizers," *IEEE Transactions on Information Theory*, vol. 51, no. 5, pp. 1739–1755, 2005.